Matpar--Parallel Extensions to Matlab

Paul Springer
High Performance Computing Group, Section 385
Paul.Springer@jpl.nasa.gov
(818) 393-3014

Matpar Introduction

Background

Matpar Architecture

Matpar Design

Current Status

Timings

Using Matpar

Future Work

Background

Original request from Space Interferometry Mission (SIM) design engineers

- Matlab too slow on large problems (2000 x 2000 size matrices)
- Many jobs had to be run overnight

Next Generation Space Telescope (NGST) needs even greater capability

- Optical design for segmented mirror
- May use 100,000 x 1,000 size matrices

Work begun in early 1996 on parallel extensions to Matlab, called Matpar

Design Goals

Use parallelism to speed only the slowest operations

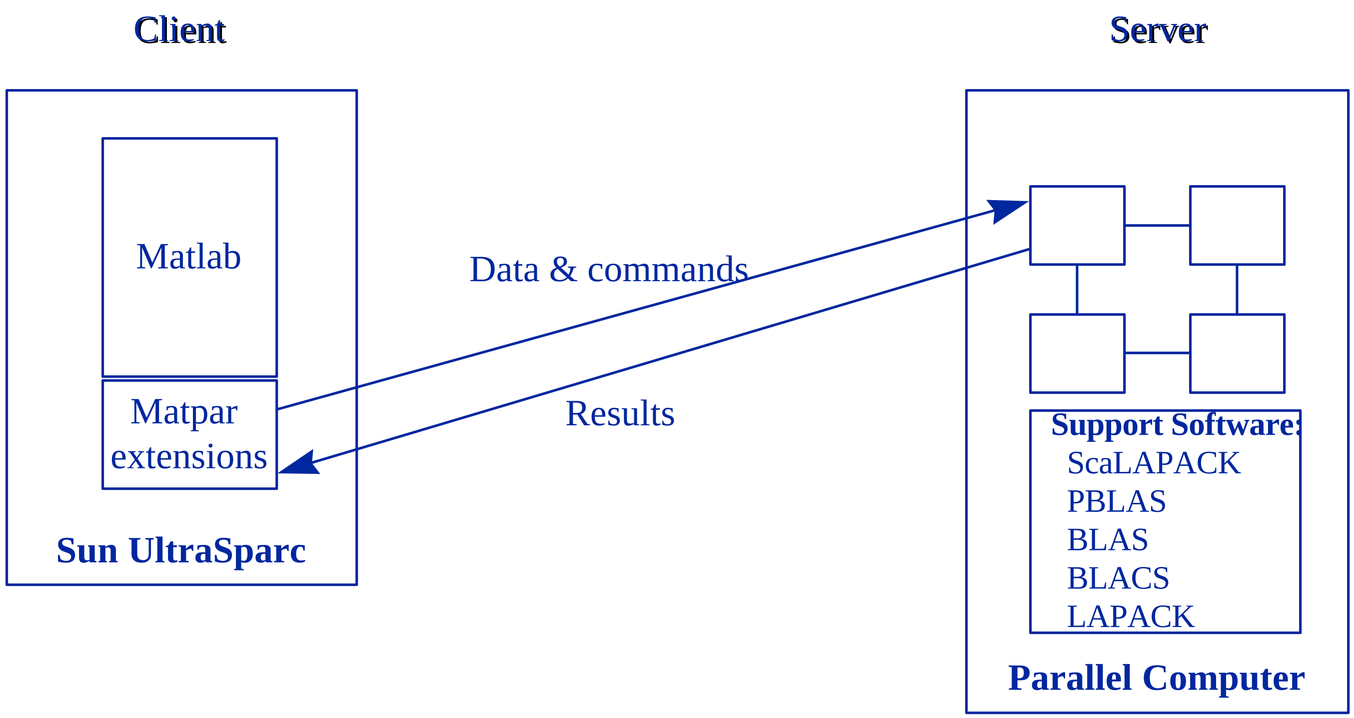
- Don’t make a parallel Matlab equivalent
- Goal: at least 1 order magnitude speedup

Keep code portable for use on different MPP’s

- Use previously ported packages: ScaLAPACK, LAPACK, BLAS, BLACS, PVM

Initially port to Cray T3D and Intel Paragon

Matpar Architecture



Matpar Design

Only certain operations parallelized

Simple Matlab style function calls

- B=qr(A) becomes B=p_qr(A)
- C=A*B becomes C=p_mult(A,B)

Calls can be made seamlessly

Calls invoke .m and .mex files

- Matlab standard method of extending the language

Matpar code starts PVM session on MPP

Parallel routines invoked, results returned via PVM to Matlab

Current Status

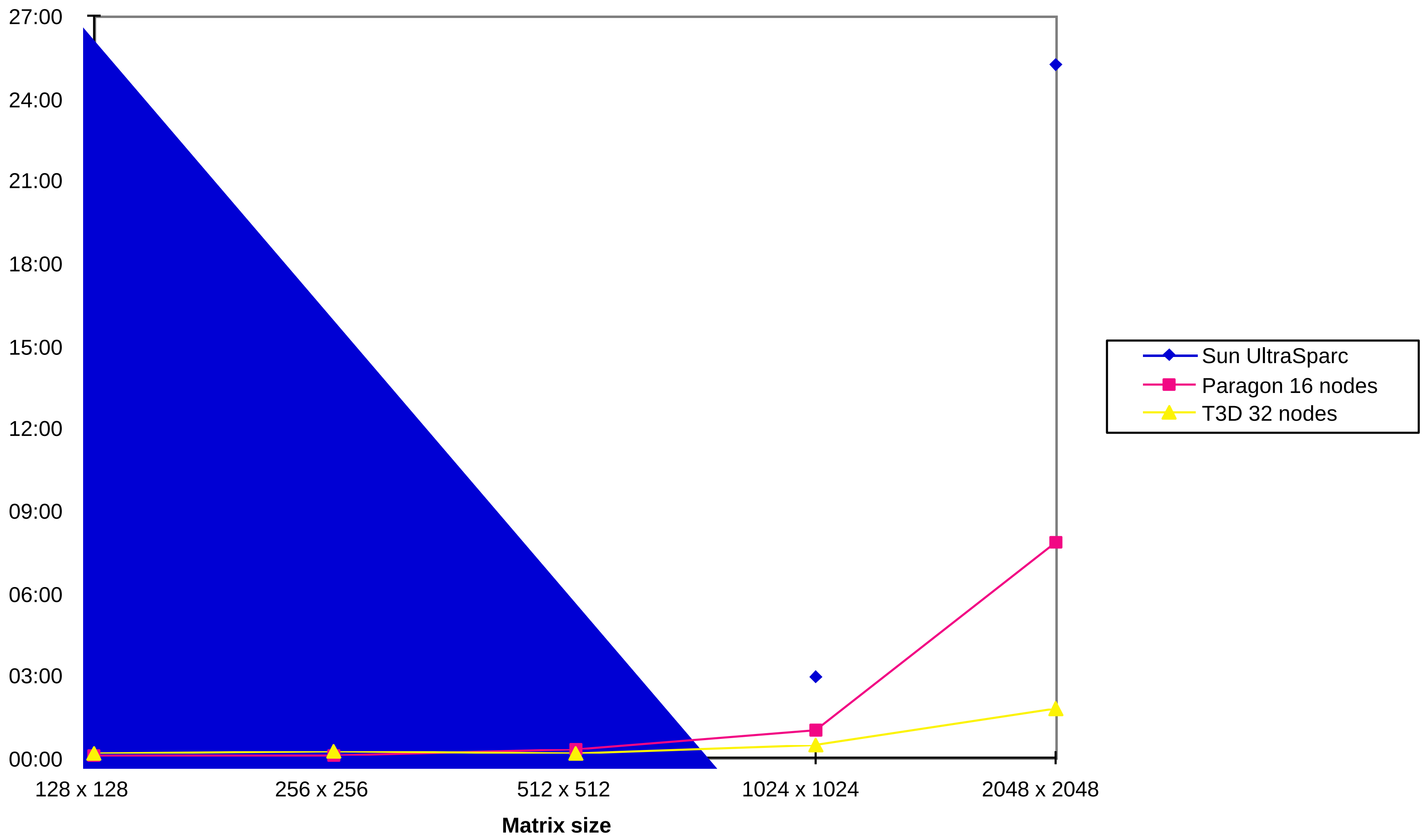
Functionality

- Matrix-matrix multiplication: p_mult(A,B)
- Calculate $A * A^T$: p_multtrans(A)
- Trace of $A * A^T$: p_trace(A,1)
- QR factorization: p_qr(A)
- LU factorization: p_lu(A)
- Solve A*X = B: p_solve(A,B)
- Frequency response calculations: p_freqresp(A,B,C,D,w)
- Bode plot calculations: p_bode(A,B,C,D,w)
- Select computer and node count: p_config(computer, node count)
- Persistence: p_persist(A,1)

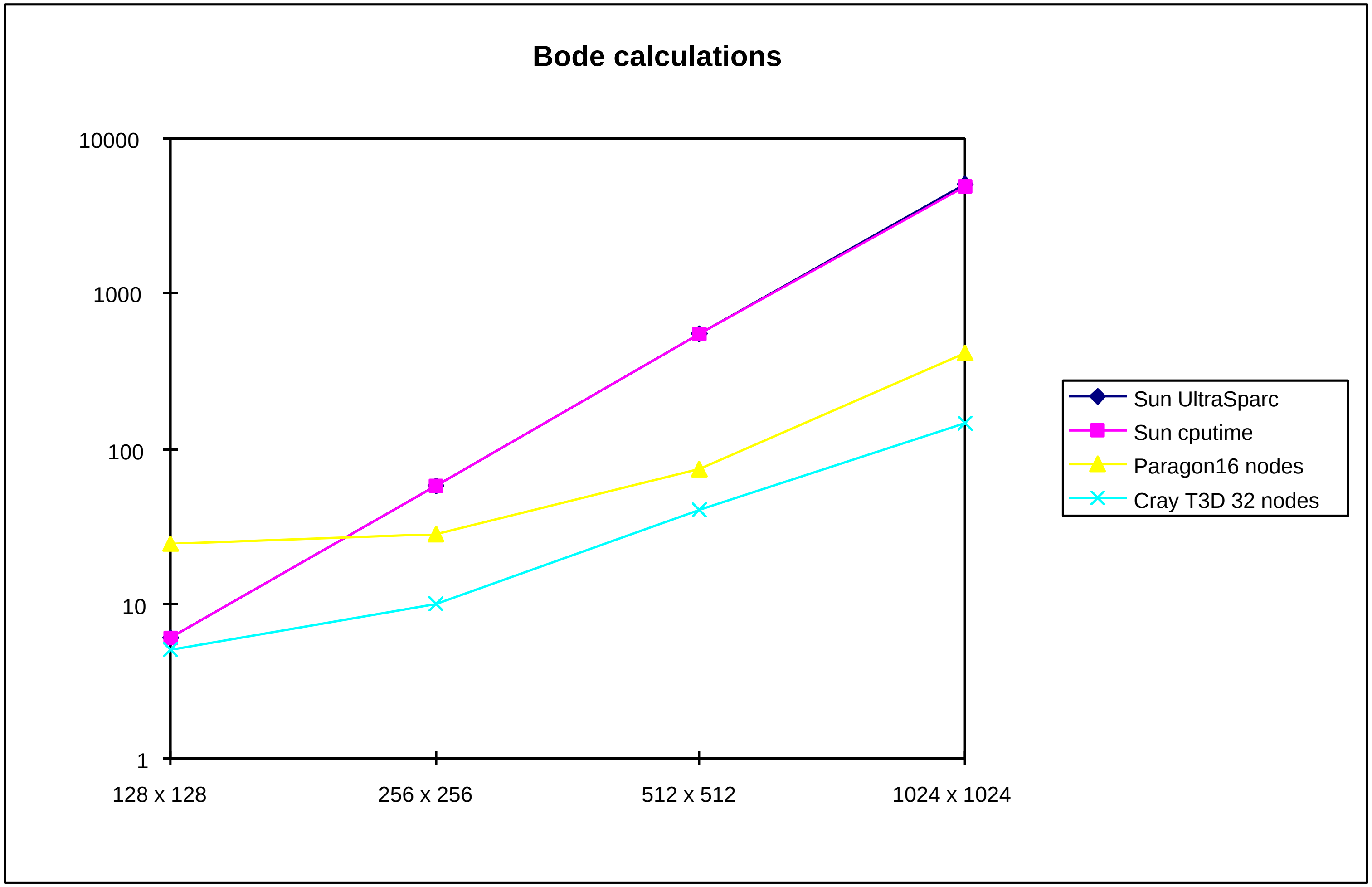
Ports

- Intel Paragon
- Cray T3D

QR Factorization Times



Timings--Bode Plot Calculations



Sample Run: Screen 1

```
hothead:/home/pls/matlab/client>[57] pvmd
7f000001:acd5
Password (cosmos.jpl.nasa.gov:pls):
```

Sample Run: Screen 2

```
hothead:/home/pls/matlab/client>[38] matlab
>> echo
>> p_config('cosmos', 32)
>> jpbode
load jmelody/mpi
tic
[mag,phas]=p_bode(full(A),full(B),full(C),D,2*pi*f);
Adding host cosmos to virtual machine . . .
Starting 32 matpar tasks
in 4 rows and 8 columns
Sending to 655360
G2=mag.*(cos(phas/180*pi)+sqrt(-1)*sin(phas/180*pi));
TIME2=toc;
>> TIME2

TIME2 =

  102.7309

>> TIME

TIME =

   1.6827e+03
```

Usability Considerations

One time setup procedure for each user per workstation and MPP

T3D faster but less flexible

Matrix data distribution

- Incompatibilities between operations
- p_persist() defaults to even distribution
- Porting to ScaLAPACK 1.5 will fix this

Data transport bottlenecks

- Need high speed Ethernet connection and interface cards (UltraSparcs)

Memory

- About 1024 x 1024 limit for Bode on Paragon & T3D
- Convex SPP2000 port should do better

Future Work

Port to Convex SPP2000

Get user feedback

Prepare for release 1.1

Completely handle complex matrices

Update to use ScaLAPACK 1.5 (matrix redistribution)

Port to Beowulf (PC cluster) and network of Suns